

---

# **Simulation Tests of the Optimization Method of Hopfield and Tank Using Neural Networks**

---

Russell A. Paielli, Ames Research Center, Moffett Field, California

November 1988



National Aeronautics and  
Space Administration

**Ames Research Center**  
Moffett Field, California 94035



## SUMMARY

The method proposed by Hopfield and Tank for using the Hopfield neural network with continuous-valued neurons to solve the traveling salesman problem (TSP) is tested by simulation. Several researchers have apparently been unable to successfully repeat the numerical simulation documented by Hopfield and Tank. However, it appears that the reason for those difficulties is that a key parameter value is reported erroneously (by four orders of magnitude) in the original paper. When a reasonable value is used for that parameter, the network performs generally as claimed. Additionally, a new method of using feedback to control the input bias currents to the amplifiers is proposed and successfully tested. This eliminates the need to set the input currents by trial and error.

## INTRODUCTION

The neural network approach to computation is based on highly parallel, analog architecture. The neural networks proposed by Hopfield and Tank would ultimately be constructed as integrated packages of resistors, capacitors, and operational amplifiers. The operational amplifiers, which constitute the artificial "neurons," would be operated well beyond their linear ranges and into saturation. The potential applications are in pattern recognition, complex motion control, and other problems which have proven to be either too computationally intensive or simply not practical for conventional digital computers. Neural networks have the potential for rapidly solving some very difficult problems that require excessive amounts of time on digital computers.

Hopfield and Tank (ref. 1) have shown that neural networks can be applied to certain optimization problems. In particular, they propose that neural networks may be useful in solving combinatorial optimization problems very rapidly, provided that one does not require the absolute optimal solution but only a reasonably good one. They show how to "map" a particular combinatorial optimization problem, the traveling salesman problem (TSP), onto the network. The traveling salesman problem may be stated as follows: given  $n$  city locations, plan a tour for a salesman such that each city is visited exactly once, with the salesman ultimately returning to the starting city, in the minimum possible distance.

The mapping of the traveling salesman problem onto the neural network consists of the specification of an energy function of, and a physical interpretation of, the output state of the network. The network seeks the minimum energy state, which is interpreted according to predetermined rules as a particular solution of the TSP.

## Hopfield Network

In the most general configuration of the Hopfield neural network, the output of each operational amplifier ("neuron") would be connected to the input of every other amplifier through some effective conductance, forming a matrix of conductances, or a "connection matrix". The input of each amplifier is also connected through a parallel RC network to ground. The input/output relationship of each operational amplifier approximates a sigmoid function (fig. 1). The gain of each amplifier

is defined as the slope at zero input. For convenience, the outputs of the amplifiers are scaled to lie between 0 and 1 over the full range.

The network is a nonlinear dynamical system which may be described with state-space notation. The state vector of the system,  $x$ , represents the inputs to the amplifiers and the output vector,  $y$ , represents the outputs of the amplifiers. The matrix of conductances,  $T$ , and the input bias currents to the amplifiers,  $I$ , determine the trajectory of the system according to the differential equation

$$\dot{x} = -x/\tau + Ty + I \quad (1)$$

where

$$y = g(x) \quad (2)$$

with  $g(x)$  representing the sigmoid function and  $\tau$  being the time constant of the network, which is the time constant of the identical RC networks connected between the input of each amplifier and ground.

The network seeks a local minimum of an energy function given by

$$E = -\frac{1}{2} y^T T y - y^T I \quad (3)$$

By proper selection of  $T$  and  $I$ , the energy function can be designed to effectively map various optimization problems onto the network. For example, for an unconstrained quadratic function minimization problem where  $E$  represents the function to be minimized, the elements of  $T$  are given by

$$T_{ij} = -\frac{\partial^2 E}{\partial y_i \partial y_j} \quad (4)$$

or, more succinctly, the  $T$  matrix may be expressed as

$$T = -\frac{\partial^2 E}{\partial y^2} \quad (5)$$

and  $I$  is given by

$$I = -\frac{\partial E}{\partial y} - T y \quad (6)$$

Substituting equation (6) into equation (1), we have

$$\dot{x} = -x/\tau - \frac{\partial E}{\partial y} \quad (7)$$

$$= -x/\tau - \frac{\partial x}{\partial y} \frac{\partial E}{\partial x} \quad (8)$$

Thus it is apparent that the behavior of the network is very similar to that of a gradient-descent minimization algorithm. The effective gradient-descent rate,  $\partial x / \partial y$ , is a time varying diagonal matrix whose components (as can be seen from the sigmoid function of figure 1) increase in magnitude as  $x$  moves away from the origin of the state space. Note that, since the network effectively performs a gradient-search, only local minima can be found.

# TRAVELING SALESMAN PROBLEM

The traveling salesman problem is a classic problem in combinatorial optimization which may be stated as follows: given  $n$  city locations, plan a tour for a salesman such that each city is visited exactly once, with the salesman ultimately returning to the starting city, in the minimum possible distance. This problem is interesting because it is both simple to state and, for a large number of cities, difficult to solve. In principal, the solution is simple: exhaustive search. However, the search space grows combinatorially with the number of cities and quickly becomes impractical to search for even the most powerful computers. The number of possible tours is  $(n-1)!/2$ .

Hopfield and Tank have devised a way of mapping the traveling salesman problem tsp onto the neural network (ref. 1). For  $n$  cities, the state vector  $x$  and the output vector  $y$  are each composed of  $n \times n$  elements which, although treated as vectors of dimension  $n^2$  for purposes of state-space notation, are actually interpreted as  $n \times n$  arrays. The energy function is designed to force the equilibrium output array of the network to the form of a permutation matrix, which has a single 1 in each row and column and 0 elsewhere. The order of each city in the tour is then read off as the column containing the 1 in the row corresponding to that particular city. For example, the output array

$$\left. \begin{array}{ccccc} 0 & 1 & 0 & 0 & a \\ 0 & 0 & 0 & 1 & b \\ 1 & 0 & 0 & 0 & c \\ 0 & 0 & 1 & 0 & d \end{array} \right\} \text{city}$$

would be interpreted as the city-sequence c-a-d-b.

Each element of the output array  $y$  is confined to the range from 0 to 1 by virtue of the sigmoid function. By defining the energy function in a particular way, it is shown that the elements of the output array can be forced to values of 0 or 1 at equilibrium. That is, the output can be forced to move to a *corner* of the  $n$ -dimensional "hypercube" which constitutes the output space. For convenience, the notation  $y_{ki}$  is used to denote the element of the output array corresponding to the  $k$ -th city (row) in the  $i$ -th position (column). The energy function is given by

$$\begin{aligned} E = & A/2 \sum_i \sum_j \sum_{l \neq j} y_{ij} y_{il} \\ & + B/2 \sum_j \sum_i \sum_{i \neq k} y_{ij} y_{kj} \\ & + C/2 \left( \sum_i \sum_j y_{ij} - n_0 \right)^2 \\ & + D/2 \sum_i \sum_{k \neq i} \sum_j d_{ik} y_{ij} (y_{k,j+1} + y_{k,j-1}) \end{aligned} \quad (9)$$

where  $d_{kl}$  represents the distance between the  $k$ -th and  $l$ -th cities (for notational convenience we treat the subscripts as modulo  $n$ , i.e.,  $n+1 \rightarrow 1$ ,  $0 \rightarrow n$ ),  $n_0$  ( $\neq n$ ) is a parameter used to provide an offset to the neutral positions of the amplifiers, and  $A$ ,  $B$ ,  $C$ , and  $D$  are parameters selected by the experimenter.

The energy equation (9) translates, according to equation (5), into a connection matrix given by

$$\begin{aligned}
T_{ij,kl} = & -A\delta_{ik}(1 - \delta_{jl}) \\
& -B\delta_{jl}(1 - \delta_{ik}) \\
& -C - Dd_{ik}(\delta_{l,j+1} + \delta_{l,j-1})
\end{aligned} \tag{10}$$

(where  $\delta_{ij} = 1$  if  $i = j$  and  $\delta_{ij} = 0$  otherwise) with external input currents

$$I_{ij} = Cn_0 \tag{11}$$

The terms of equation (10) have the following interpretation. The terms with coefficients  $A$ ,  $B$ , and  $C$  provide the constraints for the traveling salesman problem in general. The term with coefficient  $A$  provides the inhibitory connection within each row, which inhibits more than one neuron from being activated in each row at equilibrium. The term with coefficient  $B$  has the same function for the columns. The term with coefficient  $C$  provides the overall excitation, which tends to make the total number of activated neurons at equilibrium equal to the number of cities. The term with coefficient  $D$  provides the information regarding the locations of the cities and is thus the only term specific to each particular problem. At equilibrium, if the network settles to a valid permutation matrix, the latter term is equal to the total tour distance.

Substituting equations (10) and (11) into equation (1) yields the following differential equation describing the state of the network.

$$\begin{aligned}
\dot{x}_{ij} = & -x_{ij}/\tau - A \sum_{l \neq j} y_{il} - B \sum_{k \neq i} y_{kj} \\
& -C \left( \sum_k \sum_l y_{kl} - n_0 \right) \\
& -D \sum_k d_{ik} (y_{k,j+1} + y_{k,j-1})
\end{aligned} \tag{12}$$

and

$$y_{ij} = 1/2 (1 + \tanh(x_{ij}/x_0)) \tag{13}$$

for all  $i$  and  $j$ . The sigmoid function of equation (13) is intended to approximate the input/output characteristics of an actual operational amplifier. The parameter  $x_0$  is the inverse of the gain of the amplifiers.

The network is initialized by setting the value of all the state variables equal, so that the sum of all the outputs is equal to the equilibrium sum  $n$ , the number of cities, then adding uniformly distributed random noise of relative magnitude 10% to each state variable. The random noise is necessary to break symmetry, because otherwise the network cannot distinguish between the  $2n$  tours of identical length and thus does not converge. Since the network is a free-running system, the initial state determines the final output state.

### Feedback Control of Input Current

As stated above,  $n_0$  is not set equal to  $n$  in equations (9), (11), and (12). This provides an offset of the neutral positions of the amplifiers. The required offset depends on the city locations

and the parameters of the network and cannot be easily predicted. Hopfield and Tank apparently selected the value of  $n_0$  by trial and error for their particular example problem. It is now proposed that the network just described be modified to have feedback control of the value of  $n_0$ , which is accomplished with feedback control of the input current.

The feedback control is intended to force the total excitation level of the network to the proper value at equilibrium. At equilibrium, the scaled sum of the outputs of all the amplifiers should equal the number of cities,  $n$ . The feedback control law simply adjusts the value of  $n_0$  according to the difference between the sum of the outputs and the number of cities, as follows:

$$\dot{n}_0 = -K \left( \sum_k \sum_l y_{kl} - n \right) \quad (14)$$

where  $K$  is a constant feedback gain. The exact initial value of  $n_0$  is not critical; it may as well be set to  $n$  or somewhat greater, perhaps  $1.5n$ .

## SIMULATION RESULTS

The simulation of the traveling salesman problem solution by Hopfield and Tank was duplicated as accurately as possible. With the exception of the time constant  $\tau$ , the parameters were set to the same values used by Hopfield and Tank. Hopfield and Tank made the statement (ref. 1) "Without loss of generality,  $\tau$  can be set to 1." That statement is apparently incorrect, and may have led one group to conclude (ref. 2), "Our simulations indicate that Hopfield and Tank were very fortunate in the limited number of TSP simulations they attempted." A more reasonable value for  $\tau$  is  $10^{-4}$ . The parameters were thus

$$A = B = 500, \quad C = 200, \quad D = 500$$

$$x_0 = 0.02, \quad \tau = 10^{-4}$$

In this evaluation, a simple first-order Euler integration scheme was used, with an integration time increment of  $\Delta t = 10^{-5}$ , which is one-tenth the time constant  $\tau$ .

To establish a baseline, the traveling salesman problem was also solved by using the trivial nearest-city approach. One city was chosen as the starting point, and the "salesman" simply continued going to the nearest city which had not yet been visited until all cities had been visited. The initial city is arbitrary as far as the closed-path distance is concerned, but the choice of initial city affects the result of the nearest-city algorithm. Therefore, each city was tried as the initial city and the best result was recorded. The execution time of the complete algorithm was thus of order  $n^2$ , which grows neither combinatorially nor exponentially with the number of cities.

The network configuration parameters given above were used on five randomly located sets of 10 and 15 cities. For each set of cities, 20 different random initializations of the network were tried. The results are summarized in tables 1 and 2 for the 10-city and the 15-city problems, respectively. The tables show, for each randomly located set of cities, the number of trials resulting in valid solutions (permutation matrices), the number resulting in solutions as good as or better than the nearest-city solution, the normalized tour distance of the best solution (normalized with the nearest-city tour distance), and the normalized average tour distance for all the valid solutions.

Plots of the best neural network solution of the 20 trials for each random city set are shown in figures 2 and 3 for the 10-city and the 15-city problems, respectively. For purposes of comparison,

the best nearest-city solution is shown to the left of each corresponding neural network solution. The individual plots are ordered according to the tour numbers in the tables.

According to these results, the neural-network approach cannot be depended on to reliably provide even a valid solution of the traveling salesman problem: for the 10-city case, 78% of the trials resulted in valid solutions; for the 15-city case, 72% were valid. Of those valid solutions, 51% were better than the best nearest-city solution for the 10-city case and 28% were better for the 15-city case. For all of the 10-city cases and all but one of the 15-city cases, the best of the 20 neural-network solutions was better than the best nearest-city solution. However, the average of the valid solutions was slightly worse than the best nearest-city solution for all but one of the city sets.

Of course, the network performance could possibly be improved by tuning the parameters more carefully, but that would be very time consuming and computationally intensive so it was not attempted.

An attempt was made to solve randomly-generated 20-city traveling salesman problems, but the network configuration with the parameters given above was completely unsuccessful at converging to valid tours. No extensive effort was made to find a better set of parameters. A brief effort was made to incorporate slowly increasing amplifier gains, as was done by Hopfield and Tank for the 30-city problem, but this was not successful. Thus a serious question remains as to how useful the neural network method is for the large-scale traveling salesman problem. Unfortunately, the simulation becomes so computationally intensive for the case with many cities that the potential payoff may not justify the development effort involved in retuning or reconfiguring the network. This could change if an actual analog-hardware network becomes available.

## CONCLUSIONS

The use of the Hopfield neural network with continuous-valued neurons to solve optimization problems is equivalent to a gradient-descent algorithm which minimizes a quadratic cost function. Thus only local minima can be found.

The method proposed by Hopfield and Tank for using the network to solve the traveling salesman problem has been tested by simulation. An error in one of the parameters reported in the Hopfield and Tank paper appears to be the source of convergence difficulties reported by several researchers. Apparently the network time constant should be approximately  $10^{-4}$  rather than 1. Once that error is corrected, the network performs generally as claimed.

A new method of feeding back the total excitation level of the network to control the input bias currents to the amplifiers has been tested and found to work very well. The method eliminates the need to use trial and error to set the input currents.

Since there is a  $2n$  redundancy in the solution of the traveling salesman problem (starting city and direction of tour is arbitrary) a partially random initialization of the neurons is required. Otherwise the network cannot break the symmetry and hence does not converge at all. When the parameters given by Hopfield and Tank are used, not all random initializations lead to a useful solution. In some cases, the network does not converge to a valid tour at all; in other cases, the network converges to a valid tour which is worse than the simple nearest-city solution. However,



if several different random initializations are tried, some are likely to result in a valid tour with a shorter tour distance than that resulting from the nearest-city solution. This was shown by simulation for the 10- and 15-city traveling salesman problem.

## REFERENCES

1. Hopfield, J. J.; Tank, D. W.: "Neural" Computation of Decisions in Optimization Problems. *Biological Cybernetics*, vol. 52, pp. 141-152, 1985.
2. Wilson, G. V.; Pawley, G. S.: On the Stability of the Traveling Salesman Problem Algorithm of Hopfield and Tank. *Biological Cybernetics*, vol. 58, pp. 63-70, 1988.

Table 1: SUMMARY OF RESULTS OF SIMULATED NEURAL NETWORK SOLUTION OF TRAVELING SALESMAN PROBLEM WITH 10 RANDOMLY LOCATED CITIES.

Tour number	Number trials	Valid solutions	Good <sup>a</sup> solutions	Normalized <sup>b</sup> minimum	Normalized <sup>b</sup> average
1	20	19	13	0.9596	0.9912
2	20	14	4	0.9999	1.0810
3	20	20	15	0.9228	1.0117
4	20	7	5	0.9628	1.1241
5	20	18	14	0.9374	1.0418
Total	100	78	51	—	—

<sup>a</sup> as good as or better than best nearest-city solution

<sup>b</sup> normalized with best nearest-city tour distance

Table 2: SUMMARY OF RESULTS OF SIMULATED NEURAL NETWORK SOLUTION OF TRAVELING SALESMAN PROBLEM WITH 15 RANDOMLY LOCATED CITIES.

Tour number	Number trials	Valid solutions	Good <sup>a</sup> solutions	Normalized <sup>b</sup> minimum	Normalized <sup>b</sup> average
1	20	20	3	0.9999	1.0919
2	20	17	0	1.0460	1.1601
3	20	8	7	0.8945	1.0930
4	20	13	10	0.9414	1.0570
5	20	14	8	0.9582	1.0675
total	100	72	28	—	—

<sup>a</sup> as good as or better than best nearest-city solution

<sup>b</sup> normalized with best nearest-city tour distance

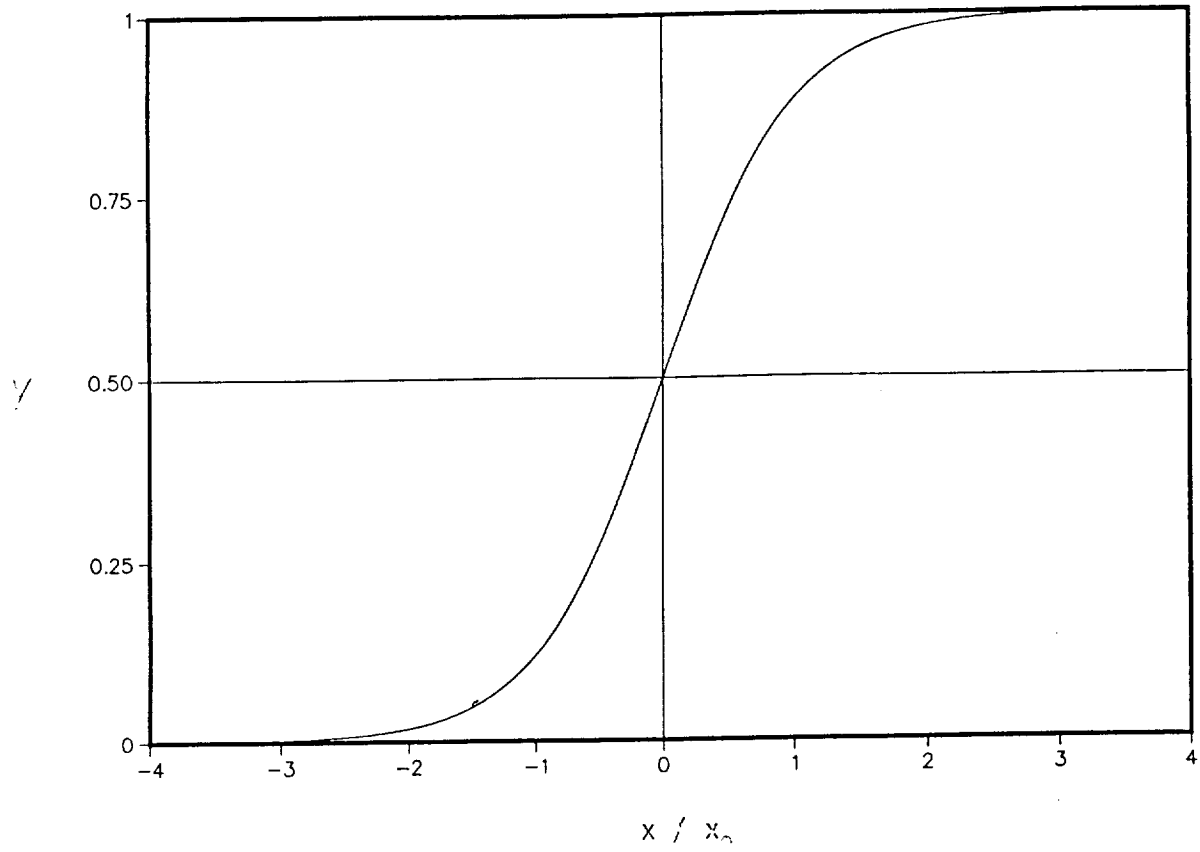


Figure 1: Operational amplifier input-output relationship: the sigmoid function.

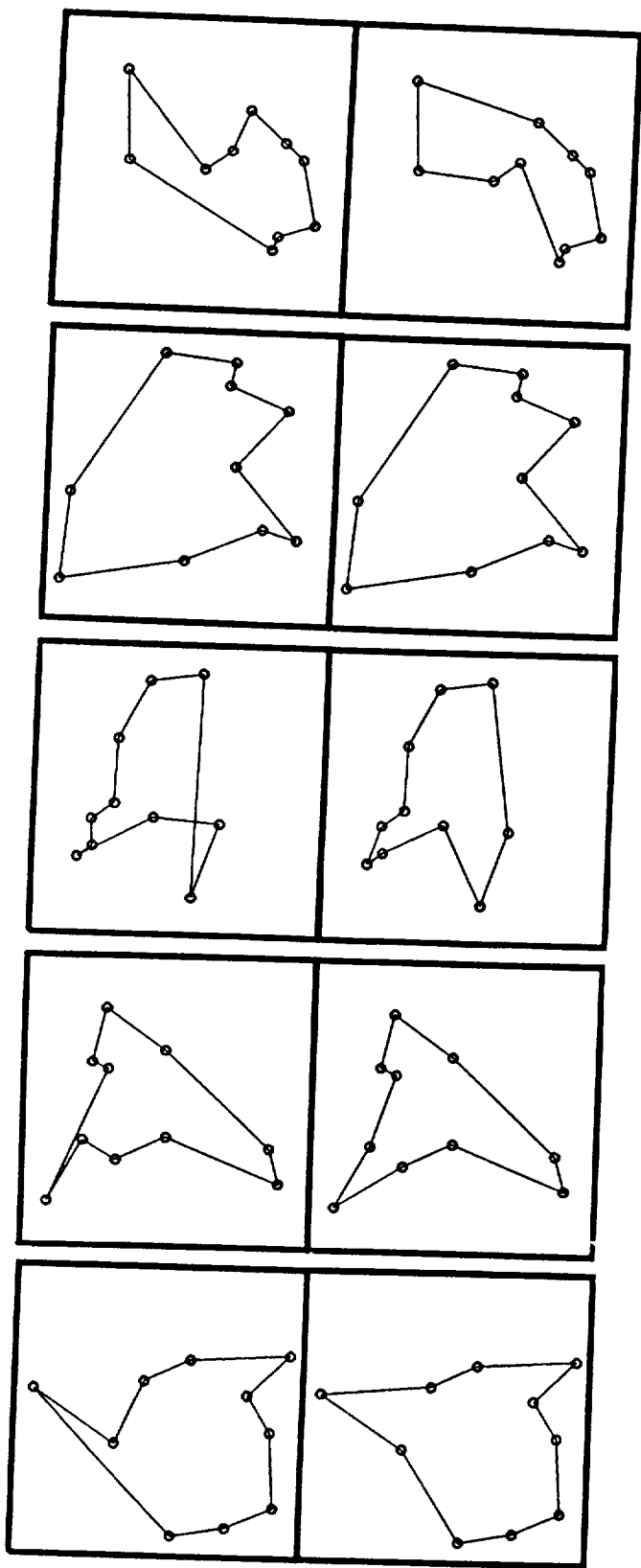


Figure 2: Right column: best neural network solutions of the 10-city traveling salesman problem. Left column: corresponding best nearest-city solution.

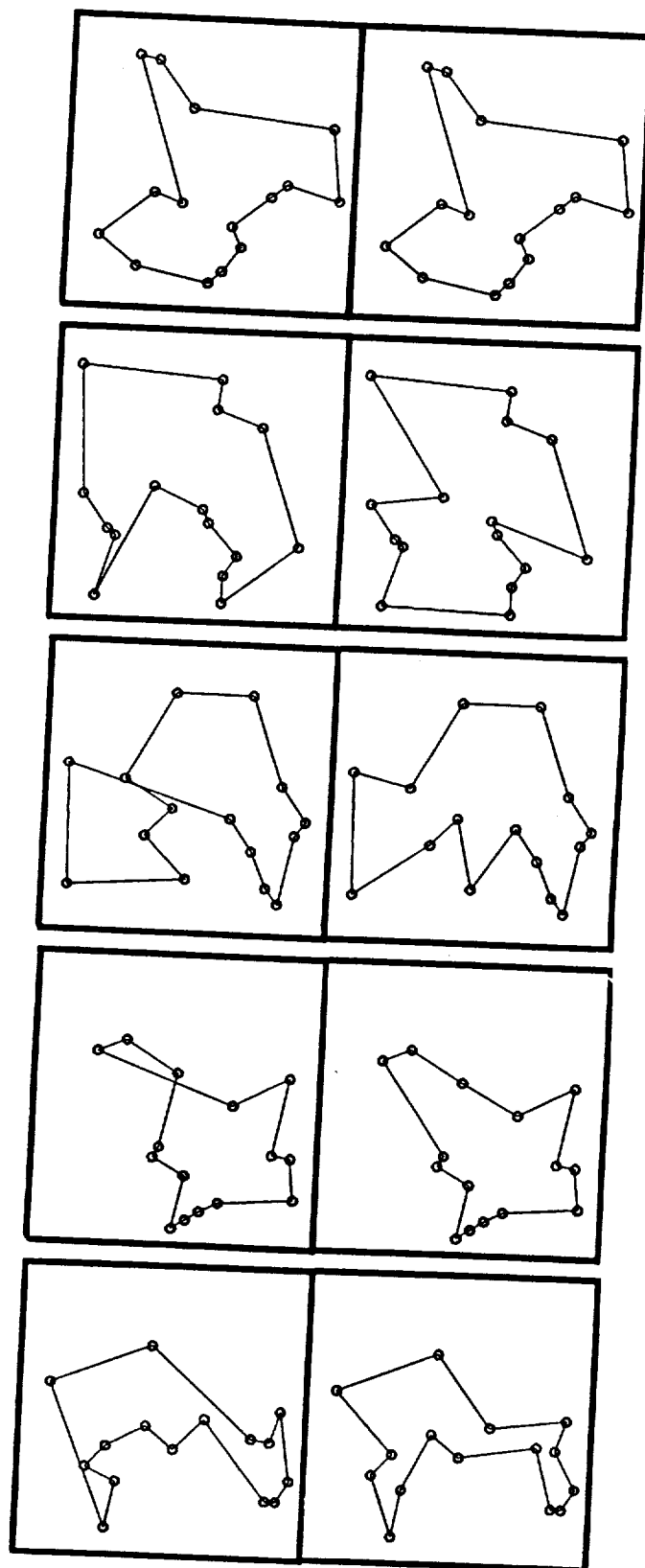


Figure 3: Right column: best neural network solutions of the 15-city traveling salesman problem. Left column: corresponding best nearest-city solution.



## Report Documentation Page

1. Report No. <b>NASA TM-101047</b>		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle <b>Simulation Tests of the Optimization Method of Hopfield and Tank Using Neural Networks</b>				5. Report Date <b>November 1988</b>	
				6. Performing Organization Code	
7. Author(s) <b>Russell A. Paielli</b>				8. Performing Organization Report No. <b>A-88275</b>	
				10. Work Unit No. <b>505-66-51</b>	
9. Performing Organization Name and Address <b>Ames Research Center Moffett Field, CA 94035</b>				11. Contract or Grant No.	
				13. Type of Report and Period Covered <b>Technical Memorandum</b>	
12. Sponsoring Agency Name and Address <b>National Aeronautics and Space Administration Washington, DC 20546-0001</b>				14. Sponsoring Agency Code	
15. Supplementary Notes <b>Point of Contact: Russell A. Paielli, Ames Research Center, MS 210-9, Moffett Field, CA 94035 (415) 694-5454 or FTS 464-5454</b>					
16. Abstract <p>The method proposed by Hopfield and Tank for using the Hopfield neural network with continuous-valued neurons to solve the traveling salesman problem (TSP) is tested by simulation. Several researchers have apparently been unable to successfully repeat the numerical simulation documented by Hopfield and Tank. However, as suggested to the author by Adams, it appears that the reason for those difficulties is that a key parameter value is reported erroneously (by four orders of magnitude) in the original paper. When a reasonable value is used for that parameter, the network performs generally as claimed. Additionally, a new method of using feedback to control the input bias currents to the amplifiers is proposed and successfully tested. This eliminates the need to set the input currents by trial and error.</p>					
17. Key Words (Suggested by Author(s)) <b>Combinatorial Optimization Traveling salesman problem Neural network, Hopfield</b>			18. Distribution Statement <b>Unclassified – Unlimited</b>  <b>Subject Category – 63</b>		
19. Security Classif. (of this report) <b>Unclassified</b>		20. Security Classif. (of this page) <b>Unclassified</b>		21. No. of pages <b>12</b>	
				22. Price <b>A02</b>	





